REMARKS/ARGUMENTS

In the above-identified Office Action, the Examiner imposed a restriction requirement by dividing the claims into two groups. Group I was drawn to "creating a new object, classified in class 717, subclass 108." Group II was drawn to "shutting down an object, classified in class 717, subclass 108." The undersigned hereby confirms that a provisional election was made to prosecute the claims of Group I, as stated by the Examiner on page 3 of the Office Action, in the first sentence of paragraph 5 thereof. Applicants hereby affirm this election.

Claims 1-14 and 23-25 were rejected as being anticipated by White (US Patent 5,603,031). The Examiner explained in paragraph 8 starting at the bottom of page 3 of the Office Action that White's patent described the Claim 1 step of "creating a new object for a new instance of the application, using an existing object of an existing instance ..." in White's abstract, and in FIGs. 4A and 4B – elements 100, 102, 120, 150, 220, 130, and associated text, FIGs. 5A and 5B – elements 120, 140, 150, 220 and associated text, FIG. 5C – elements 520, 522, 524, 526, 140, 220, 150 and associated text, FIG. 8A – elements 120, 220, 150, and associated text, FIG. 11A – element 100 and associated text, FIG. 12 – element 1270 and associated text, col. 7:25 to col. 13:34 (emphasis added to objects, agents, agent class, computers, network, place process, place class, and meeting place).

Note that although the Examiner cited to specific reference numbers in the above-quoted explanation, these reference numbers are not helpful in understanding the rejection, because the Examiner has cited **all (or almost all) of the reference numbers** disclosed in each cited figure (e.g. all reference numbers in FIGs. 4A and 4B were listed; all reference numbers in FIGs. 5A and 5B were listed; and all reference numbers in FIG. 5C were listed; and so on). Moreover, as understood by the undersigned, the Examiner has cited **over seven columns of text** from the White patent against this **single limitation** in Claim 1. As seen from the following quotation of the cited text from the White patent, the amount of text is so vast as to make it quite difficult to understand the Examiner's rejection. Moreover, the Examiner's indication of emphasis (see end of previous paragraph) was not helpful in understanding the rejection. The undersigned has

SILICON VALLEY PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

selectively emphasized certain text in the following quotation, which is discussed in greater detail below.

TEXT ASSOCIATED WITH FIGs. 4A & 4B:

FIG. 4A shows computer network 100 which includes computer systems 120A and 120B which are connected by communications link 102AB. Computer system 120A is executing place 220A and agent 150A. Agent 150A occupies place 220A as shown in FIG. 4A. Computer system 120B is executing place 220B. Hereinafter, the statement that an agent or place occupies a particular place should be interpreted as including a statement that the agent or place is executing within the computer system which contains the particular place.

Agent 150A issues an instruction to computer system 120A. In response to the instruction, agent 150A is transported to a place, e.g. place 220B, specified in the instruction. The instruction is called operation "go", and the issuance of the instruction by agent 150A is called herein performance of operation "go" by agent 150A.

Upon performance of operation "go" by agent 150A, (i) **execution of agent 150A is suspended;** (ii) agent 150A is encoded into a standardized form which preserves the execution state of agent 150A; (iii) the standardized form of agent 150A is transported to computer system 120B; (iv) agent 150A, including the preserved execution state, is decoded from the standardized form; and (v) execution of agent 150A is resumed within computer system 120B. After performance of operation "go" by agent 150A, agent 150A no longer occupies place 220A and is no longer executing within computer system 120A; instead, agent 150A occupies place 220B and is executing within computer system 120B (FIG. 4B). By enabling an agent to travel to a remote computer system in the midst of the agent's execution, the agent is free to travel to data which the agent is configured to access.

...

In one embodiment, a ticket 1306 (FIG. 4A) controls the transportation of agent 150A and specifies place 220B as the destination place. As shown in FIG. 4A, agent 150A contains ticket 1306. Ticket 1306 defines the trip taken by agent 150A from place 220A to place 220B. In one embodiment, ticket 1306 specifies (i) the destination place of the trip, (ii) the "way" or pathway and "means" agent 150A is to take to the destination place, (iii) the maximum amount of time within which the trip must be completed before the trip is aborted and (iv) the amount of resources agent 150A asks to use at the destination place. By

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

specifying the amount of resources agent 150A is permitted to use at place 220B, place 220B can determine whether agent 150A requires more resources than place 220B is configured to provide. In such a situation, place 220B can deny ingress to agent 150A. Thus, using a ticket as described more completely below, an agent can completely define a pending trip between a first place and a second place.

...

TEXT ASSOCIATED WITH FIGs 5A-5C:

To assist in the understanding and visualization of the various aspects and embodiments of the present invention, various representational and relational conventions are used in the drawings. Representational and relational conventions used herein are represented in FIGS. 5A, 5B and 5C. Computer system 120A (FIG. 5A) is executing place 220A, place 220X, place 220Y, agent 150A, agent 150X and agent 150Y. Computer system 120A also contains objects 140A, 140B and 140X in a memory, e.g., mass memory or main memory (neither shown), of computer system 120A. Agent 150A and 150X occupy place 220A; place 220Y occupies place 220X; and agent 150Y occupies place 220Y. Agent 150A owns objects 140A and 140B, and agent 150X owns object 140X. The relationships of occupancy and ownership are discussed in greater detail below. Briefly, occupancy of places provides various levels and types of security, and ownership determines which objects travel with an agent which performs operation "go".

FIG. 5B is an alternative and equivalent representation of the various relationships represented in, and described above with respect to, FIG. 5A. While FIG. 5A accurately represents, e.g., that agent 150X occupies place 220A and contains object 140X, the form of FIG. 5A is less suitable for representing more complex relationships. Therefore, the tree structure of FIG. 5B is used in the drawings to represent more complex relationships in illustrating the various computer instructions of various embodiments of the present invention.

The tree structure of FIG. 5B should not be confused with the class hierarchy tree which is shown in part in FIG. 5C. FIG. 5C represents the class relationships of various items represented in FIGS. 5A and 5B. Objects 140A, 140B and 140X are members of class "Object" which is represented by class object 520. Membership in a class is discussed in greater detail in the Glossary of Terms above and is represented by a dashed line between the class and the object which is a member of the class. Class object 522 represents class "Process", which is a subclass

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

of class "Object". Places 220X, 220Y and 220A are members of class "Place", which is a subclass of class "Process" and is represented by class object 524. Agents 150A, 150X and 150Y are members of class "Agent", which is a subclass of class "Process" and is represented by class object 526. Thus, places 220X, 220Y and 220A and agents 150A, 150X and 150Y are all members of class "Process" and are therefore computer processes. Objects 140A, 140B and 140X are not members of class "Process" and are therefore not computer processes.

TEXT ASSOCIATED WITH FIG 8A:

Meeting place 220B (FIG. 8A), which is executing in computer system 120B, provides a means for agents 150A and 150B, which occupy meeting place 220B, to communicate and share information in a meeting as indicated by arrows A and B. Petition 3106 within agent 150A defines and controls the meeting between agents 150A and 150B.

...

TEXT ASSOCIATED WITH FIG 11A:

Objects constructed of the object-oriented computer instruction set are executed by an engine, e.g., engine 132A (FIG. 11A). Engine 132A has a communication infrastructure 132A-CI, a program portion 132A-P, and a data portion 132A-D. Data representing the state of the various objects executed by engine 132A, e.g., object 140A, are stored in data portion 132A-D of engine 132A. Data portion 132A-D of engine 132A is memory space within memory 117A (FIG. 10) reserved as work space for engine 132A and is generally inaccessible from the perspective of other processes on computer system 120A, e.g. operating system 131A and user application 133A.

As discussed above, an engine effectuates execution of processes and objects. Program portion 132A-P (FIG. 11A) of engine 132A includes computer instructions which effectuate execution of the objects represented in data portion 132A-D. The computer instructions which are combined to form program portion 132A-P can be of a known computer language. For example, the computer software attached as Microfiche Appendix G is a program portion of an engine constructed in accordance with the principles of the present invention and is constructed in accordance with the C++ programming language.

Communication infrastructure 132A-CI of engine 132A includes computer instructions which transport data between engines

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

dispersed throughout network 100. The computer software attached as Microfiche Appendix H, which is a part of this disclosure and is herein incorporated by reference in its entirety, is a communication infrastructure of an engine constructed in accordance with the principles of this invention. Many aspects of the communication infrastructure of an engine are described in greater detail in Appendix D, which is a part of this disclosure and which is incorporated herein by reference in its entirety.

FIG. 11B is an alternative and equivalent representation of the computer network of FIG. 11A.

As indicated by FIG. 11A, engine 132B of computer system 120B is configured in a manner that is directly analogous to the configuration of engine 132A.

...

TEXT ASSOCIATED WITH FIG 12:

Diagram 1270 (FIG. 12) illustrates the class relations of the classes of which agent 150A is a member. Agent 150A is a member of class "Agent" as agent 150A is shown to be contained within domain 1272 which represents class "Agent". Since class "Agent" is abstract, agent 150A is also a member of one or more subclasses (not shown) of class "Agent".

Domain 1272 is completely contained within domain 1274 which represents class "Process". Agent 150A therefore inherits attributes "brand", "permit" and "privateClasses", which are defined by class "Process". Additionally, all members of class "Agent" are also members of class "Process". Thus, as described above, class "Agent" is a subclass of class "Process".

Domain 1274 is completely contained within domain 1276 which represents class "Object". Agent 150A therefore inherits attributes "class" and "size" which are defined by class "Object". Additionally, all members of class "Process", including members of class "Agent", are also members of class "Object". Thus, as described above, class "Process" is a subclass of class "Object".

Domain 1274 representing class "Process" is contained within domain 1278, which represents class "Named". Connection 1278A shows that domain 1274 is contained within domain 1278 while accurately representing that domain 1276, representing class "Object", is not contained within domain 1278 and that domain 1278 is not contained within domain 1276. Class "Named", represented by domain 1278, is a mix-in class. As domain 1274 is contained within domain 1278, agent 150A is contained within domain 1278 and is therefore a member of mix-in class "Named".

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

Mix-in class "Named" defines attribute "name" which is included in agent 150A.

Domain 1280 which represents mix-in class "Referenced" contains domain 1276, which represents class "Object", as indicated by connection 1280A. As domain 1276 is contained within domain 1280, agent 150A is contained within domain 1280 and is therefore a member of mix-in class "Referenced". Mix-in class "Referenced" defines attribute "isProtected" which is included in agent 150A.

COLUMN 7 LINE 24 TO COLUMN 13 LINE 34:

According to the principles of this invention, a set of interpreted, object-oriented computer instructions is used to create novel computer processes that are executed in a distributed computer system. A particular process that utilizes the set of computer instructions is activated by an engine that is executing within the distributed computer system. The engine interprets and effectuates the instructions of the particular process within the distributed computer system.

Each engine in the distributed computer system interprets the instructions that define a process uniformly. In other words, the instructions which comprise a process and which are interpreted by a first engine do not depend on the particular configuration of the computer system within which the first engine is executing. The instructions, therefore, can be moved to and interpreted by a second engine, even if the first and second engines are executing within two separate computer systems whose operating systems and hardware are otherwise generally incompatible. The interpreted instructions are implemented by using interfaces to communication, storage, computing and other subsystems of the computer system within which the engine is executing. These interfaces, which collectively form part of one embodiment of the present invention, are described in Appendix C, which is a part of this disclosure and is incorporated in its entirety herein by reference.

Two or more engines are interconnected to form a network. The network is a universe within which computer processes of this invention travel. In one embodiment, the network encompasses computer systems that would normally be considered clients of, and not part of, other networks. For example, one embodiment of the network of this invention encompasses the workstations which are connected by the network. The term "workstation" is used herein to describe a computer system which is used for purposes other than the transportation of information to other computer systems. The engines of this invention are interconnected so that

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

engines are capable of moving computer processes among themselves.

To transport a particular computer process, the **computer process is suspended** and the execution state of the computer process is preserved. The instructions of the computer process, the preserved execution state, and objects owned by the computer process are packaged, or "encoded", to generate a string of data that is configured so that the string of data can be transported by all standard means of communication used to form a network. In one embodiment, the string of data is transported between engines as specified by the protocol described in Appendix D, which is a part of this disclosure and is incorporated herein in its entirety by reference.

Once transported to a destination computer system of the network, the string of data is decoded to generate a computer process within the destination computer system. The decoded computer process includes those objects encoded as described above and has the preserved execution state. The destination computer system resumes execution of the computer process. The instructions of the computer process are executed by the destination computer system to perform complex operations, including defining, creating and manipulating data objects and interacting with other computer processes executed by the destination computer system. As computer processes are uniformly interpretted throughout the network and are encoded and decoded for transport between computer systems, the computer processes of this invention provide a new level of functionality and versatility in intercomputer communication.

In one embodiment, two classes built into the set of computer instructions of this invention are an agent class and a place class. Instructions formed using the agent class are interpreted by an engine to form an "agent process", sometimes referred to herein simply as "an agent". An agent is an active object in that an engine initiates execution of an agent upon creation of the agent. The agent class provides instructions which enable an agent to (i) examine and modify itself, (ii) transport itself from a first place process, which is described more completely below, in the network to a second place process in the network, and (iii) interact with other agents found at the second place process. The first and second place processes can execute within two separate computer systems of the network. Thus, an agent can travel from a first computer system to a second computer system.

An agent can contain information which is carried with the agent from the first place process to the second place process. Additionally, the extensibility, generality and functionality of the set of computer instructions discussed more completely below provide

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

tremendous flexibility and control in determining, according to the instructions which comprise an agent, how and to what destination the agent, and the information contained therein, travels.

The power of agents is counterbalanced by "permits". A permit limits the particular capabilities of a particular agent on particular occasions. The permit of an agent specifies which of several of the operations defined for the agent class can or cannot be performed by the agent. The permit further limits the amount of processing resources the agent can consume and the time at which the agent expires. Additionally, the permit specifies the priority of execution of the agent relative to other agents.

Instructions formed using the place class are interpreted by an engine to form a "place process", sometimes referred to herein simply as "a place". A place is also an active object, and the place class provides instructions which enable a place to examine and modify itself and to serve as a venue for agents and a context in which agents can interact. Agents each occupy a respective single place. Additionally, each place can occupy a single other place.

Places provide a degree of privacy and security for agents. For example, an agent, which is configured to avoid contact with other agents, can occupy a place which is not generally known to other agents, or which denies ingress to other agents. Conversely, an agent, which is configured to provide services publicly to a large number of agents, can occupy a place which is widely known to other agents and which grants ingress to other agents.

Agents cannot interact at a distance; i.e., no two agents can interact unless both occupy the same place. To interact with a second agent occupying a second place, a first agent, which occupies a first place, issues a command causing the first agent to be transported to the second place as discussed below with respect to the "go" operation. While both the first and second agents occupy the second place, the first agent can interact with the second agent.

Thus, the processes of this invention are a novel implementation of "remote programming," and not the more familiar "remote procedure calling" paradigm. Remote programming improves upon remote procedure calling by (i) enabling **processes to interact without communicating across network** communications media and (ii) improving the performance of the interactions between processes by eliminating communication across network communications media which often has high-latency.

The movement of an agent process, from a first place process in a first computer system to a second place process either in the first

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

computer system or in a second computer system is called a trip. The agent initiates the trip by using a "go" operation, which is defined in the agent class. The agent controls the movement either through the network or within the computer system by creating and submitting, as an argument to the "go" operation, a ticket which defines the trip. In one embodiment, the ticket specifies the place to which the agent is to travel, the "way" by which the agent is to travel, the amount of time in which the trip must be completed, and an indication of the urgency of the trip, i.e., the priority of the trip relative to other trips by other agents that may be concurrently scheduled.

The ticket can identify the destination place by specifying the address, name, class or any combination of the address, name and class of a place to which the agent is to travel. The destination of the trip is any place of the specified address, name and/or class which grants the agent ingress within the time permitted by the ticket.

In the "go" operation, the agent is moved from a first place to a second place as follows: (i) **the agent is suspended** and the execution state of the first agent is preserved; (ii) the agent is represented in a standardized form, i.e., an octet string, which includes representation of the agent's preserved execution state; (iii) the standardized form is transported from the first place to the second place, potentially involving the transfer of the standard form from a first computer to a second computer; (iv) the agent at the second place is formed from the standardized form, including the execution state represented in the standardized form; and (v) interpretation of the agent is resumed, the agent initially having the execution state represented in the standardized form.

As mentioned above, the set of computer instructions of the present invention is object-oriented. Therefore, all objects formed according to the present invention are organized into classes. All classes in the present invention are represented by data objects which can be moved along with an agent. Thus, classes, which are not defined within a place, can nevertheless be used by an agent travelling to the place simply by the agent defining the classes and transporting the corresponding class objects to the place.

Every object in one embodiment of the present invention is owned by a process, i.e., either an agent or a place. When an agent travels from a first place to a second place, all objects owned by the agent are effectively moved to the second place along with the agent. However, as discussed below, objects which are equivalent to objects already occupying the second place are not transported in one embodiment of the present invention. In addition to the objects owned by the agent, all class objects defining classes of

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

which those objects are members are moved along with the agent to the second place. A class object is an object constructed in accordance with the set of computer instructions described below and in Appendix A which represents a class of objects.

Thus, if an object owned by an agent travelling to a second place is a member of a class not defined in the second place, the object can still travel with the agent as the agent also carries to the second place class objects defining the classes of which the object is a member. In the prior art, a process which travelled from a first computer to a second computer could only process data objects which belonged to classes defined on the second computer. Class definitions were not typically transported with migrating processes in the prior art. As new classes can be formed within a first place and members of those new classes are free to travel to other places for which those new classes are not defined, the present invention provides agents a level of extensibility, mobility and generality not found in the prior art.

In the present invention, many classes of objects are defined at multiple places and therefore do not need to be moved to such places. Class objects and any other objects, which are likely to be found at many places are made interchangeable in one embodiment of the present invention.

Interchangeable objects each have a digest. An interchangeable object, which is owned by an agent that is travelling from a first place to a second place, does not travel with the agent. Rather, the digest of the interchangeable object is moved with the agent to the second place. When the agent arrives at the second place, interchangeable objects in the computer system which contains the second place are examined to determine whether any of the interchangeable objects has a digest equal to the digest transported with the agent. If such an interchangeable object is found in the computer system which contains the second place, the interchangeable object is substituted for the interchangeable object, which was left behind at the first place.

If, however, no such interchangeable object is found at the second place, the interchangeable object left behind at the first place is moved to the second place. In this way, the movement of objects across network communications media is avoided when equivalent objects are present at the destination place. In particular, as class objects are interchangeable, an agent travelling to a second place causes the movement to the second place of only those class objects defining classes which are not defined within the second place. Avoiding unnecessary movement of class objects through the network makes practical the movement of objects owned by an agent to places which contain

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

no definition of one or more classes to which those objects belong.

In one embodiment of the present invention, classes are identified by citations. In a computer network having many computer systems supplied by different vendors, various versions of the disclosed instruction set can be implemented on various computer systems to which and from which agent processes can travel. A citation is an object which is used to identify classes or objects which are forward- or backward-compatible with one another. Therefore, an instruction requiring use of an object of a first class, or the first class itself, can use an object of a second class, or the second class itself, depending on the particular requirements of the instruction, if the second class is backward-compatible with the first class.

An agent, occupying a first place, is also capable of creating one or more clones of the agent and moving each clone to a respective place. A clone is an agent process which is the result of duplicating an existing agent process. An agent initiates and controls the creation of one or more clones, and the movement of each clone to a respective place process by performance of operation "send." The agent specifies the number of clones to be created and defines the corresponding trips by creating one or more tickets which are supplied as arguments to operation "send". Each ticket defines a trip to be taken by a respective clone of the agent. The number of tickets supplied by the agent defines the number of clones created.

Once a clone of the agent is moved to a second place, the clone initially has the execution state of the agent at the time the clone was created. Thus, when the clone arrives at a second place, the clone continues to execute so as to simulate the movement of the agent to the second place as described above with respect to operation "go."

In the prior art, a first process, i.e., a head process, created limb processes which had the same interface, i.e., could generally perform the same operations, as the head process. Limb processes did not act except as directed by the head process and could move to remote computer systems only at the direction of the head process.

In the present invention, however, a clone is autonomous and is not controlled by the agent which created the clone. For example, the clone can be configured to ignore all attempts by the agent to arrange a meeting, which is discussed below, whereby the agent and the clone can interact. The agent must attempt to interact with a clone of the agent in the same way the agent attempts to interact with any other agent, as discussed below. As the clone is

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

autonomous, the clone occupying the second place can travel to a third place by performance of operation "go" without being so instructed by the agent and even without the consent of the agent.

As limb processes of the prior art were not active and acted only at the direction of a head process, the head process and a remote limb process necessarily interacted across network communications media. The paradigm of this prior art system is therefore more closely related to remote procedure calling. In contrast, clones of agents formed in accordance with the present invention are active and autonomous and embody all of the instructions which form the agent. Therefore, **no interaction is required (in fact, no interaction is allowed) across network communications media.** Therefore, the paradigm of the present invention is more closely related to remote programming. The present invention therefore represents a significant increase in generality over the prior art.

Increases in efficiency in one embodiment of the present invention are realized in performance of operation "send" by deferring cloning of an agent as long as possible. For example, a first clone and a second clone are to be sent to a first place and a second place, respectively. Suppose that the agent is executing within a first computer system, that the first place is executing within a second computer system, and that the second place is executing within a third computer system. Suppose further that in travelling to the first and second places, the first and second clones must travel through an engine executing on a fourth computer system. In such a case, a single clone is formed and transported to the engine executing within the fourth computer system. Thus, only a single clone is created within the engine interpreting the original agent thereby saving space within that engine and only a single clone is transported across network communications media to the fourth computer system thereby saving time in transporting clones to the fourth computer system. The engine executing within the fourth computer system forms from the single clone the first and second clones and transports the first and second clones to the first and second places, respectively.

As an agent can own objects which account for a substantial majority of the size of the agent, e.g., a rasterized graphical image such as a facsimile transmission or digitized sound, avoiding sending several copies of such large objects, each copy owned by a respective clone, to a single engine saves substantial time and expense.

A first agent, occupying a place, can initiate a meeting between the first agent and a second agent occupying the place. During such a meeting, the first agent can transfer to and receive from the second agent data in the form of objects, and the second

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

agent process can transfer to and receive from the first agent data in the form of objects.

The present invention represents a significant improvement over a prior art system which teaches the posting of messages on a virtual bulletin board by a first process. The messages are then "read" by the intended recipient process. In the prior art system, a first process gives to a second process a reference to the first process by posting the reference on the virtual bulletin board. However, since there is no mechanism for simultaneous exchange of references, the second process has access to the first process before giving to the first process a reference to the second process. Thus, there is no mechanism to prevent "malicious" processes from gaining access to other processes without granting to the other processes access to themselves.

The present invention represents an improved method of establishing contact between two processes such that a first process cannot obtain access to a second process without simultaneously granting to the second process access to the first process.

In one embodiment of the present invention, two agents can interact only if both agents occupy the same place and the place is a meeting place. A meeting place is a place that is a member of a class of meeting places, which is a subclass of the class of places. A first agent directs that a meeting be arranged between the first agent and a second agent by issuing an instruction directing the meeting place to arrange the meeting. The issued instruction is called operation "meet", and the first agent's issuance of the instruction is called "requesting a meeting".

The first agent supplies, as an argument to the instruction, a petition defining the meeting. The petition defines the meeting by specifying the second agent as the petitioned agent. The second agent is specified by specifying the name and/or the class of the second agent. The petition further defines the meeting by specifying the amount of time in which the meeting must be arranged or abandoned.

In arranging the meeting, the meeting place supplies to the second agent the name and class of the first agent and indicates that the first agent has issued an instruction requesting a meeting with the second agent. The second agent examines the name and class of the first agent and responds to the meeting place either accepting or rejecting the meeting with the first agent.

If the meeting is rejected, the first agent is informed that the second agent is not available. If the meeting is accepted, the meeting place gives to the second agent a reference to the first

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

> agent and gives to the first agent a reference to the second agent. With a reference to the second agent, the first agent (i) can direct the second agent to perform operations; (ii) can supply to the second agent objects as arguments; and (iii) can receive from the second agent objects as results. As the second agent has a reference to the first agent, the second agent has similar capabilities with respect to the first agent.
>
> Either the first or the second agent can terminate the meeting between the two by issuing an appropriate command. The meeting place, in performing an operation "part" in response to the issued command, voids any references to the second agent contained within the first agent and voids any references to the first agent contained within the second agent, thereby terminating interaction between the two agents.
>
> The present invention represents a significant improvement over the prior art as a first agent cannot gain access to a second agent unless the second agent agrees or without granting to the second agent access to the first agent. Additionally, since **two agents cannot interact unless both occupy the same meeting place**, intermediate levels of security are available to agents. For example, an agent can protect itself from other agents by occupying a meeting place whose location is not widely known by other agents. Inversely, an agent, which is designed to be highly visible, may occupy a well-known meeting place. No such mechanism is available in the prior art.

The above-quoted text is so vast that it makes the Examiner's rejection difficult to understand. Specifically, the Examiner has not said where in the above-quoted text does White teach "creating a new object for a new instance of the application, using an existing object of an existing instance ..." as recited in Claim 1. For example, it is unclear as to whether Claim 1's instance is being analogized to White's "agent" or White's "process."

Therefore, Applicants' first argument is that the Examiner has failed to make a prima facie case of rejection of this limitation of Claim 1. If the Examiner makes any prior art rejection in a future Office Action, **the Examiner is respectfully requested to provide a "pin-point" cite**, to no more than a handful of lines, for each claim limitation.

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

Moreover, the Examiner repeated, for a second time, the very same citation to White's patent (noted above) for teaching Claim 1's "setting up connectivity between the

new instance and the network connected to the plurality of computers..." See the first full paragraph on page 4 of the Office Action. The Examiner also repeated, for a third time, this very same citation in White's patent for teaching Claim 1's "starting execution of the new instance." See the second full paragraph on page 4 of the Office Action. The Examiner further repeated, for a fourth time, this very same citation for teaching Claim 1's "wherein the new instance uses the new object, and the connectivity to access a resource shared by the multiple instances." See the bottom of page 4 and top of page 5 of the Office Action. Therefore, the Examiner has not stated, as to which step in the White's method is being cited against each of these three different claim limitations. Instead, the Examiner has repeated the same citation four times, once for each of four limitations in Claim 1.

Hence, Applicants' second argument is that the Examiner has failed to make a prima facie case of anticipation of the three additional limitations of Claim 1 namely "setting up connectivity", "starting execution" and using "the connectivity to access a resource shared by multiple instances." Specifically, the Examiner failed to identify each of the four claim limitations individually in a pin-point manner within the above-quoted text. Repeated use of the same citation against each of four different limitations of Claim 1 makes the **Office Action prima facie defective**. In the absence of a one-to-one correspondence between each limitation of Claim 1 and a pin-point cite in the White patent, it is quite difficult to respond to the Office Action.

Therefore, if any claim is rejected in a future Office Action, the Applicants request the Examiner to provide a different cite for each claim limitation. Such an individual pin point citation for each limitation will avoid piecemeal examination. See MPEP §707.07(g).

Applicants' third argument is that White fails to disclose or suggest use of an existing agent, when creating a new agent. Specifically, White states that his process is **suspended when being transported**. See White's patent at column 7, line 66. column 9 lines 47-48, and at column 22 line 16. When suspended, White's process does not execute. In contrast, originally-filed Claim 1 states, in the very first limitation, that it is an

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

existing instance which is used in creating the new instance, and that the existing instance is one of multiple instances. And the multiple instances are executing, as stated in the preamble of Claim 1, and therefore Claim 1's existing instance is executing. Support for this argument is found throughout the originally-filed application, including, for example, page 5 at lines 5-6, page 6 at line 2, page 6 at line 10, page 9 at lines 22-25, page 10 at lines 10-13, page 10 at lines 19-20, page 18 at lines 13 and 14, page 19 at line 27, and page 20 at lines 2-5. There is no disclosure or suggestion by White that when his agent is executing it can be copied or moved.

Applicants' fourth argument is that White fails to disclose or suggest setting up connectivity between an agent or process and the network. Specifically, White explicitly states that his **agents cannot interact at a distance**, that no two agents can interact unless both occupy the same place. See White's patent at column 9, lines 9-10. In order to interact, one of White's agents must be transported to the other agent's place. Only when the two agents are in the same place can they interact. White further states that his processes interact without communicating across network communications media. See White's patent at column 9 at lines 21-25. Hence, White fails to disclose or suggest the second limitation of Claim 1, namely "setting up connectivity between the new instance and the network."

Applicants' fifth argument is that White fails to disclose or suggest use of network connectivity to access a shared resource. Specifically, as noted in the fourth argument, White explicitly states that his agents cannot interact at a distance. Hence, White's agent cannot use "connectivity to access a resource shared by multiple instances" as required in a further limitation of Claim 1.

In view of the above arguments, Applicants submit that White does not disclose each and every limitation of Claim 1. In order for White to anticipate Claim 1, the "identical invention must be shown in as complete detail as is contained in the ...claim." **Richardson v. Suzuki Motor Co.,** 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989); MPEP § 2131. Since White fails to disclose several limitations of Claim 1, Applicants

SILICON VALLEY PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

respectfully request the Examiner to withdraw the prior art rejection of Claim 1. Claims 2-14 depend from Claim 1 and are patentable for at least the same reasons as Claim 1.

In the above-identified Office Action, Claims 2-14 were rejected for the same reasons as Claim 1. The Examiner did not provide any specificity in citation to White's patent. Instead, the Examiner repeatedly relied on the "same figures and text" as Claim 1. See the Office Action's paragraphs 9-12 on page 5, paragraphs 13-17 on page 6 and paragraphs 18-21 on page 7. Applicants respectfully traverse these remarks by the Examiner as being an "omnibus" rejection which makes this **Office Action prima facie defective**. This defect again arises from the Examiner's failure to provide different pin-point cites for each dependent claim.

While the rejection of each of Claims 2-14 is traversed in the previous paragraph, following are a few examples of the Examiner's failure to establish anticipation of the dependent claims. For example, in rejecting Claim 2 in paragraph 9 on page 5 of the Office Action, the Examiner stated that White's above-quoted text discloses renaming of a copy of an object by using a name of the new instance. However, the undersigned has found nowhere in the entirety of the above-quoted text from White's patent is there a discussion of "renaming." White does disclose a "name" attribute (see column 31 lines 60-61 and column 32 lines 19-30), but there is no indication whatsoever that "renaming" of any kind is to be done. Claim 2 is believed to be patentable for at least this additional reason.

As yet another example, in rejecting Claim 7 the Examiner stated in paragraph 14 on page 6 of the Office Action, that White's above-quoted text discloses displaying a list of computers and receiving from a user a selection. However, the undersigned has found nowhere in the entirety of the above-quoted text from White's patent is there a discussion of displaying anything to a user. In fact the only discussion about user in the above-quoted text from White's patent is about a workspace that is inaccessible to a user application (see column 29 lines 17-20), which teaches away from the limitations in Claim 7. Claim 7 is therefore believed to be patentable for at least this additional reason.

SILICON VALLEY
PATENT GROUP LLP
2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

The Examiner rejected Claim 7 in the alternative as being obvious over White. Specifically, at page 8, in paragraph 25 of the above-identified Office Action, the Examiner stated that White fails to explicitly disclose displaying a list and receiving from a user a selection. So, in this paragraph **the Examiner contradicts the Examiner's own rejection** in paragraph 6 of the Office Action.

Applicants respectfully traverse the Examiner's official notice taken in the middle of paragraph 25 on page 8 of this Office Action. In this context, Applicants respectfully draw the Examiner's attention to the following evidentiary requirement to be met: "If Applicant Challenges a Factual Assertion as Not Properly Officially Noticed or not Properly Based Upon Common Knowledge, the Examiner Must Support the Finding With Adequate Evidence." See MPEP 2144.03.

Even assuming the Examiner supplies a prior art document that teaches generating a list of network mappings, Applicants submit that there is no motivation or suggestion to cause White's agent to transfer to a computer selected by a user. The Examiner's statement that one "would have been motivated ... to give the user the ability to select the computer in a network where a cloned object is to be created and executed" has no basis in the prior art. Applicants respectfully traverse this statement by the Examiner, and request the Examiner to supply a prior art reference for the motivation. Again, see MPEP 2144.03.

Even assuming that the Examiner provides two references as requested in the previous two paragraphs, Applicants submit that the proposed combination of a list of network mappings in Windows NT 4.0 and White's patent is improper and contrary to the precepts of M.P.E.P. § 2143.01 which requires that the proposed modification cannot change the principle of operation of a reference. The Examiner cannot, without further explanation, state that White's user application is accessing a workspace which is **explicitly described** by White as being inaccessible. The Examiner must provide a prior art basis for overcoming White's contrary teaching.

SILICON VALLEY PATENT GROUP LLP
2350 Mission College Blvd.
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210

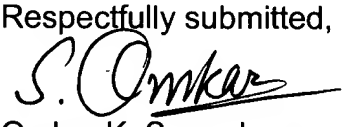
In the above-identified Office Action, at pages 8 and 9 in paragraphs 26-29, Claims 8-11 were rejected for the same reasons as Claim 1. The Examiner did not provide any additional reasoning or specificity in citation to White's patent. Once again, the Examiner repeatedly relied on the "same figures and text" as Claim 1. Applicants again traverse these remarks by the Examiner as forming an "omnibus" rejection.

New Claims 26-29 are added. Support for the new claims is found throughout the application, including, for example, for Claim 26 see page 15 lines 28 and 29, for Claim 27 see page 12 lines 29-30, for Claim 28 see page 12 lines 21-25, and for Claim 29 see page 14 lines 10-11.

Applicants believe all Claims 1-14 and 23-29 are patentable over the teachings of White. Therefore, Applicants respectfully request allowance of all pending claims. Should the Examiner have any questions concerning this response, the Examiner is invited to call the undersigned at (408) 982-8200, ext. 3.

Via Express Mail Label No.
EV 581 853 464 US

Respectfully submitted,



Omkar K. Suryadevara
Attorney for Applicants
Reg. No. 36,320

SILICON VALLEY
PATENT GROUP LLP

2350 Mission College Blvd
Suite 360
Santa Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210